Docket No. 4574-4001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s)	:	Adrian Peck & Ian Harvey	Group Art Unit: TBA Examiner: TBA
Serial No.	:	10/665,744	
Filed	:	September 19, 2003	
For	:	SECURE TRANSMISSION OF DATA WITHIN A DISTRIBUTED COMPUTER SYSTEM	

CERTIFICATE OF MAILING (37 C.F.R. §1.8(a))

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

I hereby certify that the attached:

1. Claim to Convention Priority
2. Certified Priority Document
3. Return Receipt Postcard

along with any paper(s) referred to as being attached or enclosed and this Certificate of Mailing are being deposited with the United States Postal Service on date shown below with sufficient postage as first-class mail in an envelope addressed to the: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Respectfully submitted,
MORGAN & FINNEGAN, L.L.P.

Dated: February 20, 2004 By: _____
Walter G. Hanchuuk
Reg. No. 35,179

Correspondence Address:
MORGAN & FINNEGAN, L.L.P.
345 Park Avenue
New York, NY 10154-0053
(212) 758-4800 Telephone
(212) 751-6849 Facsimile

Docket No. 4574-4001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Adrian Peck and Ian Harvey

Group Art Unit: TBA

Serial No.: 10/665,744

Examiner: TBA

Filed: September 19, 2003

For: SECURE TRANSMISSION OF DATA WITHIN A DISTRIBUTED COMPUTER SYSTEM

CLAIM TO CONVENTION PRIORITY

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313
Sir:

In the matter of the above-identified application and under the provisions of 35 U.S.C. §119 and 37 C.F.R. §1.55, applicant(s) claim(s) the benefit of the following prior application(s):

Application(s) filed in:	United Kingdom
In the name of:	nCipher Corporation Limited
Serial No(s):	0317742.5
Filing Date(s):	July 29, 2003

☒ Pursuant to the Claim to Priority, applicant(s) submit(s) a duly certified copy of said foreign application.

☐ A duly certified copy of said foreign application is in the file of application Serial No. _____, filed _____.

Respectfully submitted,
MORGAN & FINNEGAN, L.L.P.

Dated: February 20, 2004

By: _____
Walter G. Hanchuk
Registration No. 35,179

Correspondence Address:
MORGAN & FINNEGAN, L.L.P.
345 Park Avenue
New York, NY 10154-0053
(212) 758-4800 Telephone
(212) 751-6849 Facsimile

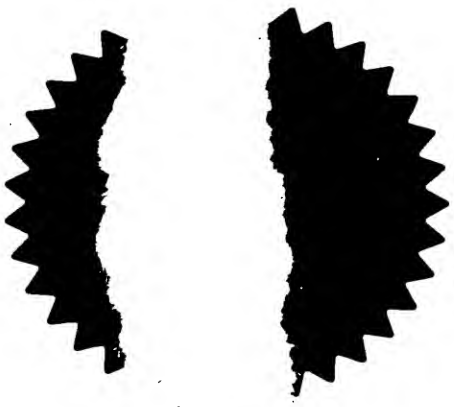
The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.



Signed

Dated 23 October 2003

Patents Form 1/77

Patents Act 1977
(Rule 16)

The Patent Office

30JUL03 E826297-4 D00056
P01/7700 0.00-0317742.5

1/77

THE PATENT OFFICE
29 JUL 2003
RECEIVED BY HAND

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

The Patent Office

Cardiff Road
Newport
South Wales
NP9 1RH

1. Your reference P35362GB/MNM

2. Patent application number *(The Patent Office will fill in this part)* 0317742.5

29 JUL 2003

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

nCipher Corporation Limited
Jupiter House
Station Road
Cambridge
Cambridgeshire CB1 2JD

7249840002

Patents ADP number *(if you know it)*

If the applicant is a corporate body, give the country/state of its incorporation: Great Britain

4. Title of the invention

Secure Transmission of Data within a Distributed Computer System

5. Name of your agent *(if you have one)*

"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*

Kilburn & Strode
20 Red Lion Street
London
WC1R 4PJ

Patents ADP number *(if you know it)* 125001 ✓

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing *(day / month / year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*

a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
See note (d))

NO

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	28
Claim(s)	11
Abstract	0
Drawing(s)	14 + 14

10. If you are also filing any of the following, state how many against each item.

Priority documents	
Translations of priority documents	
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	
Request for preliminary examination and search *(Patents Form 9/77)*	
Request for substantive examination *(Patents Form 10/77)*	
Any other documents *(please specify)*	

11. I/We request the grant of a patent on the basis of this application.

Signature MN Maggs Date 29/7/07.

12. Name and daytime telephone number of person to contact in the United Kingdom

MAGGS, Michael Norman
Tel: 020 7539 4200

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.

b) Write your answers in capital letters using black ink or you may type them.

c) If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.

d) If you have answered 'Yes' Patents Form 7/77 will need to be filed.

e) Once you have filled in the form you must remember to sign and date it.

f) For details of the fee and ways to pay please contact the Patent Office.

Secure Transmission of Data within a Distributed Computer System

The present invention relates to the secure transmission of data within a distributed computer system, particularly although not exclusively to facilitate secure communication with client devices on a network.

While the invention may be utilised in all types of network-aware devices, it is expected to find particular application in fields such as network cards, wireless LAN cards, voice over IP telephones, modems, routers, switches, hubs, TV set-top boxes, mobile telephones and the like.

Traditionally, suppliers of network devices, and particularly consumer devices, have tended to accord a relatively low priority to the implementation of security by means of cryptography including encryption and digital signatures. Although cryptographic systems can in some cases be added on to pre-existing networks, either by means of software or by means of purpose-designed hardware security modules, such solutions are often expensive and complex to implement. Some progress has been made in consumer markets by allowing the protection of cryptographic keys in smart cards or in hardware boxes, but both of these approaches are of rather limited scope. A more recent trend is toward the increasing "hardening" of client devices, with cryptography being built in as a fundamental part of the messaging protocols to be used. Unfortunately, suitably powerful embedded processors capable of carrying out the full range of cryptographic functions, including key generation, are by no means cheap and can be large and power-hungry, which discourages their use within inexpensive consumer products.

Many current systems rely on Public Key Infrastructure (PKI) in order to verify the integrity (authenticity of origin) of data transmitted across the network.

the distributor, further encrypting the encrypted data according to the protocol to generate a secure message, and transmitting the message to the client, and at the client: confirming the integrity of the transmission by decrypting the message according to the protocol, and recovering the data by decrypting the encrypted data using the secret.

The invention, in its various forms, allows us:

(a) To define a form of Secure Processor and associated functions and protocols, which is designed to minimize its physical and computations resources by offloading functions to another Secure Processor that services it.

(b) To define a framework of networked Secure Processors of that type that provides centralized storage, processing and distribution.

(c) To define such a framework to support Policies that execute in Secure Processors that automate the enforcement of rules on the processing and distribution.

(d) To define such a framework that allows separation of trust, including confidentiality and integrity, between parts of the framework.

The invention further extends to a computer system including a plurality of clients, each having a security module as previously defined and a provider arranged to send messages as required, to the said clients.

1.1 Secure Processor Definitions

For the purpose of definition, a Secure Processor 100 of known generic type, as shown in figure 1, comprises:

- a CPU 10;

- Persistent Secure Data 11, which is information stored long-term, for example, using technologies such as ROM and/or EEPROM;

- Transient Secure Data 12, which stores data only when the CPU is active, for example, using technologies such as RAM, and in particular such data cannot be expected to persist over power downs.

A Secure Processor has one or more Secure Program(s) 13, which are stored in its Persistent Secure Data 11 and can execute on its CPU 10, when it can store and access Persistent Secure Data 11 and Transient Secure Data 12, and, potentially, other resources, such as a secure clock (not shown).

An Insecure Computer can run Insecure Programs that can communicate with a Secure Processor.

In contrast to an Insecure Program, the integrity of a Secure Program and the integrity and confidentiality of its execution on its Secure Computer is regarded as trustworthy.

The Secure Computer 100 is typically a dedicated hardware module, such as an HSM (Hardware Security Module), a smart card, specialist embedded device, a

ensuring the freshness or appropriateness of data by sending it only when the Micro Secure Processor needs it; transferring data that has the effect of delegating a command that would normally be done by the Micro Secure Processor to SP2.

1.3 Secure Transfer Protocol

One Secure Processor ('SP1') can communicate with another Secure Processor ('SP2') using a 'Secure Transfer Protocol', which allows SP1 to request some data from SP2. The connection between SP1 and SP2 may involve one or more Insecure Computers. SP1 could be a Secure Processor or a Micro Secure Processor.

The Secure Transfer Protocol is designed so that the insecure parts of this system do not compromise the security of the exchange.

The connections between SP1 and SP2 can be permanent or intermittent. It can use wired or wireless networks, or a mixture of both.

1.4 Data

The Data transmitted by the Secure Transfer Protocol could contain:

a) Key Data, which describes a cryptographic key ('Key') that SP1 can use securely;

b) Program Data, which is a program that could run securely on SP1, or be loaded to run on an Insecure Computer that is directly associated with SP1;

a) a symmetric key ('Kei-secret') or;

b) an asymmetric key pair comprising 'Kei-private' and 'Kei-public'.

An entity can authenticate itself, or enforce integrity, to another Secure Processor using:

a) Kei-secret to form a Message Authentication Code (MAC) or;

b) using Kei-private to make a digital signature.

1.6 Seat of Trust Processes

As shown in figure 1, a 'Seat of Trust Process' 14 can generate the Seat of Trust 19 from one or more 'Seat of Trust Constant'(s) 15, each of which can be:

a) Persistent Secure Data 11 (e.g. inserted by the device manufacturer or at some stage during the sale, distribution or commissioning of the device);

b) supplied to the Secure Processor from some external source 16, for example, a person entering a Personal Identity Number ('PIN'); or from a biometric mechanism that encodes some personal property as a Seat of Trust Constant.

1.7 Simple Enrollment

Before SP2 can authenticate SP1, or rely on its statement of integrity, SP2 must receive:

or:

b) sign({Kei-public, Nonce, other data}, Kei-private)

where '{a, b}' denotes 'a concatenated with b' and 'sign(c, d)', means 'sign c using key d'

Then SP1 sends the Certificate to SP2.

4. SP2 verifies the Certificate and the value of Nonce, by

either:

a) mac(Certificate, Kei-secret)

or;

b) verify(Certificate, Kei-public)

where 'verify(a,b)' means 'verify a using key b'.

and checks that the value of the Nonce in the Certificate is that same as that created by SP2 in Step (1.). SP1 could also authenticate SP2 as well, where required (not shown).

4.1) If this succeeds, proceed to Step (5.).

4.2) If this does not succeed, the transfer terminates with an error and goes to Step (9.).

5. SP1 and SP2 exchange messages that result in a shared 'Session Key', ('Ks').

encrypt({Data, other data}, Ke-encrypt).

This double-encryption concept allows the system to make a distinction between the security involved in distributing Data (e.g. a key) and in the use of that Data. The message being passed allows the recipient to rely on the confidentiality of the data that is being distributed and, entirely separately, to rely on the integrity of the distribution process itself. Provided that the secret known to the client (SP1) is kept confidential from the distributor (SP2), the client can be sure that the confidentiality of the Data is preserved.
In one embodiment, for example, the Repository can serve up keys to micro-HSM endpoints without (except possibly fleetingly in an HSM connected to the repository) having any access to the key material.

The data is here encrypted twice:

- An inner encryption protects the data from the Repository.
- An outer encryption allows for micro-HSM revocation.

The Repository ensures that the inner encrypted blob is only sent to a micro-HSM if the micro-HSM (eg as identified by its Kei) is valid (eg it has not been revoked by the Regional Authority mentioned in section 4.2). The outer encryption makes use of temporary session keys which stops an attacker from successfully replaying old key transfer messages to attempt to load an encrypted blob onto a micro-HSM which has been revoked.

One major advantage of such an arrangement is that the privacy of the key needed to read the original data can be maintained entirely outside the system. This key can be kept securely in one place (for example in the head office of a

8. SP1 obtains Ke-decrypt by unwrapping Message2 with the corresponding unwrapping key, 'Kw-unwrap', which SP1 obtained from some out of-band-process

unwrap(decrypt (Message2, Ks), Kw-unwrap)

9. SP2 computes 'Message3'

encrypt(Encrypted Data, Ks)

and sends it to SP1

10. SP1 obtains Data, by decrypting Message3:

decrypt(decrypt (Message3, Ks), Ke-decrypt)

where 'decrypt(a, b)' denotes 'decrypt a with key b'.

Another variant is where a nested set of unwrapping keys is transferred in steps essentially similar to (7.) and (8.). This could be done recursively

1.10 Wrapping Key

One case of the protocol in (1.8), shown in figure 4, is where SP1 has an Entity Confidentiality Key (Kec):

either

a) a symmetric key, 'Kec-secret'

A Repository 50, which centralizes the storage of encrypted Data 52 within a data store 53 and may originate that Data or receive it from some out-of-band mechanism.

The Repository comprises at least one Secure Processor 51 or an Insecure Computer interfaced with at least one Secure Processor.

The Secure Program responsible for the Repository's activities is called the Guardian 54. This communicates with a Transferrer 55.

In one configuration, as shown in figure 5, the Repository and Provider are merged into one component. Alternatively, the Repository may communicate with one or more Provider(s), using the Secure Transfer Protocol.

Each Provider is responsible for serving Data on request to one or more User(s) 56, each of which requests and may use Data.

The Transferrer 55 communicates with the User 56 using the Secure Transfer Protocol.

Both the Provider and the User contain a Micro Secure Processor or a Secure Processor, typically interfaced to an Insecure Computer (not shown).

Providers allow for essentially arbitrary scaling to large numbers of Users and for physical and logical partitioning of distribution of data to multiple Users. In one preferred embodiment, the network structure may comprise three levels: a repository acting as a Secure Processor to several providers, with each provider itself acting as a Secure Processor to several Users. It will be

d) As shown in figure 7, SP2 is the Secure Processor in the Repository and SP1 is the Secure Processor in the Provider;

In case (a) and (b), the protocol is the same as Secure Transfer Protocol (1), (1.8), or Secure Transfer Protocol (2), (1.9), except that during Step (6.), the now Guardian retrieves the Data based on the RequestedName.

Typically cases (c) and (d) are used together (figure 7).

In case (c), (figure 7) the protocol is the same as Secure Transfer Protocol (1), or (2), but during Step (6.) the Transferer

either:

c1) initiates an exchange of type (d) to retrieve the Data or Encrypted Data based on the RequestedName from the User

or;

c2) retrieves Data or Encrypted Data that originated from a previous exchange of type (d), from a store under its control.

In the case of Encrypted Data, the Transferer may or may not choose to decrypt it.

3. Third Scenario

In this section we define a framework that extends the second scenario to support Policies that execute in Secure Processors, which automate the enforcement of rules on the processing and distribution.

The Policy could also describe how to determine the current version of the Data in a DataSet, or what to do when Data in a DataSet becomes out of date.

The Policy could also contain rules about the key lengths, algorithms and its access control lists, and other data, which are used as a template when creating a Key Instance; and similarly on Program Data and Other Data.

A Policy is typically signed by the object that is responsible for it, to ensure its authenticity and integrity. If so, a Policy is used only when this signature has been verified.

The Policy could be transferred as Other Data with the Secure Transfer Protocol, or some other means, to execute or be interpreted on another Secure Processor.

3.2 Region

We turn now to figure 9. A Guardian 90 partitions the objects for which it is responsible into one or more domains, called 'Region'(s) 91, such that each object exists in only one Region.

A Region 91 contains:

- one or more Entity(ies) 92 (each a secure processor) that can make requests of the associated Repository or Provider;

- zero or more Policies 93, which define the rules that a DataSet and or objects in the Region must obey;

A Relying Party can then authenticate a message that the Entity called Entity Name, which they have signed with Kei-private, using Kei-public from the Entity Name's Certificate, obtained from the Guardian.

For example, the binding of an Entity name to a kei-public is useful when defining policies, for instance it allows a security officer to specify by name which entities are allowed to use which application keys; or for linking an Entity's request for Data with a policy that, for example, charges that Entity Name's credit card; or securely records the transaction in an audit log, or to authenticate some other process.

3.4 Secure Transfer Protocol (4)

We turn now to figure 10. This instance of the protocol is the same as in Secure Transfer Protocol (3), (2.2), with these differences:-

a) During Step (4.), the Guardian or the Transferrer confirms that the Entity associated with Kei-public is still a member of the Region and that its Kei-public is still available from the Guardian.

If so, it continues with the protocol; if not, it exists to Step (9.) with an error.

b) During Step (6.), the Guardian or Transferer retrieves the Data as follows:-

6.1) Find the DataSet, 'DS1', that corresponds to the RequestedName.

6.2) Find the Policy in DS1, 'Policy1' that corresponds to selecting Data Instance from the DataSet using RequestedName.

Before the Guardian retrieves the DataSet corresponding to RN1, it does the following:-

1. Retrieve all Authority Groups that contain EN1;

2. Retrieve from these the Authority Group, AG1, that gives EN1 the right to download the DataSet associated with RN1.

3. Confirm that EN1's rights apply in the current context, for example, by running some associated Policy.

4. Only if all these steps are successful, does the Guardian retrieve the Data associated with RN1 and send it to SP1.

4.2 Region Authority Group

The concept of the Region Authority Group is illustrated in figure 12. The Region Authority Group 121 is an administrative group which may be solely responsible for creating and deleting the Region's DataSets; and for maintaining the Policies associated with a DataSet.

In the case when these Policies are signed, they are typically signed by the Region Authority Group, using 'Kgi-private'.

The Regional Authority Group 121 may be responsible for enrolling an Entity into the Region, for authorizing a new Authority Group, or adding an Entity to an Authority Group.

In practice, the Guardian obtains the entities public key via some sort of private secure route. For example, where the entity is an office telephone, it may be enrolled to the Guardian in a secure room within the IT department. Alternatively, the information needed to be passed between the Guardian and the entity to effect enrolment may be done in some other secured way, including for example via physical documentation being passed through a company's internal mail system.

4.4 Group Confidentiality

One particular aspect of an Authority Group is that it could keep its Data and/or Policies private from Entities in other Authority Groups and, in particular, from the Repository and Provider.

We call Data that is protected in this way Group Encrypted Data.

To this end, an Authority Group 141 has:

a) a Group Confidentiality Key symmetric key 142 ('Kgc-secret' for short)
or;
b) a Group Confidentiality Key Pair asymmetric key pair ('Kgc-private', Kgc-public').

c) The Repository receives Group Encrypted Data 143, which is placed there by an out-of-band process.

Group Encrypted Data is encrypted using

either:

Claims

1. A method for the secure transmission of data from a distributor to a client over a computer network, the method comprising:
   (a) encrypting the data using an encryption confidentiality key known to the client but not the distributor;
   (b) storing the encrypted data at the distributor;
   (c) generating a message by further encrypting the encrypted data using an encryption transmission key, the corresponding transmission decryption key being known to the client; and
   (d) transmitting the message to the client.

2. A method as claimed in claim 1 in which, on receipt of the message, the client confirms the authenticity of origin of the transmission by decrypting the message using the transmission key.

3. A method as claimed in claim 2 in which the client confirms the confidentiality of the data by decrypting the encrypted data using a confidentiality decryption key corresponding to the confidentiality encryption key.

4. A method as claimed in any one of claims 1 to 3 in which the data comprises or includes a cryptographic key.

5. A method as claimed in any one of claims 1 to 3 in which the data comprises or includes a program.

6. A method as claimed in any one of claims 1 to 3 in which the data comprises or includes licence or configuration information.

14. A method as claimed in claim 9 in which encrypted data held within the repository is divided into data sets, each data set being associated with a respective policy which defines how the data within the data set may be used.

15. A method as claimed in claim 14 in which data from a particular data set, when sent by the provider, is accompanied by the respective policy.

16. A method as claimed in claim 15 in which the policy is run by the provider.

17. A method as claimed in claim 14 or 15 in which the policy is run by the client.

18. A method as claimed in claim 14 in which the policy is run by the repository

19. A method as claimed in claim 9 in which a plurality of regions are defined within the repository, each region containing information on the secure computers that are permitted to make requests for or otherwise manipulate data held by the repository.

20. A method as claimed in claim 9 in which the said secure computers include that of the provider.

21. A method as claimed in claim 9 in which the said secure computers include those of the clients.

30. A method as claimed in claim 19 in which the information within each authority group, when there is more than one such group, is encrypted and is confidential from other groups.

31. A computer security module having means for receiving from a sender a message comprising twice-encrypted data, means for decrypting the message according to a protocol known to both the module and the sender, and means for further decrypting the decrypted message using a secret known to the module but not to the sender.

32. A computer system including a plurality of clients, each having a security module as claimed in claim 31, and a provider arranged to send messages, as required, to the said clients.

33. A computer system as claimed in claim 32 in which the provider includes a secure computer.

34. A computer system as claimed in claim 33 in which the secure computer within the provider includes a security module as claimed in claim 31.

35. A computer system as claimed in any one of claims 32 to 34 including a plurality of providers, and a repository arranged to send data, as required, to the said providers.

36. A computer system as claimed in claim 32 in which encrypted data is stored at the provider, and is re-encrypted prior to being sent as a message to the client.

45. A computer system as claimed in claim 35 in which a plurality of regions are defined with the repository, each region containing information on the secure computers that are permitted to make requests for or otherwise manipulate data held by the repository.

46. A computer system as claimed in claim 45 when dependent upon claim 33 in which the said secure computers include that of the provider.

47. A computer system as claimed in claim 45 when dependent on claim 2 in which the said secure computers include those of the clients.

48. A computer system as claimed in claim 45 when dependent upon claim 40 in which each region further includes a plurality of data sets.

49. A computer system as claimed in claim 45 in which each region is associated with a respective region policy which defines how the information within the region may be used.

50. A computer system as claimed in claim 45 in which each region further contains one or more authority groups, the or each group defining a set of secure computers that are permitted to carry out certain tasks.

51. A computer system as claimed in claim 50 in which a given secure computer may belong to a plurality of authority groups.

52. A computer system as claimed in claim 50 in which each region includes a region authority group which is responsible for administrative functions relating to its respective region.

(ii) recovering the data by decrypting the encrypted data using the secret.

58. A method as claimed in claim 57 in which the data comprises or includes a cryptographic key.

59. A method as claimed in claim 58 in which key management functions, for example key generation, are provided for the client by a secure external key management provider.

60. A method as claimed in claim 58 in which the client is adapted to use cryptographic keys but not to generate them, instead requesting a key as required from a secure external key management provider.

61. A method as claimed in claim 58 in which the key is used in a secure process by the client.

62. A method as claimed in claim 57 in which the data comprises or includes a program.

63. A method as claimed in claim 57 in which the data comprises or includes licence or configuration information.

64. A method as claimed in any one of claims 57 to 63 in which the secret known to the client is not known to the distributor.

65. A method as claimed in claim 64 in which the distributor generates the message by calculating

encrypt(wrap({Ke-decrypt}, Kw-wrap), Ks)

69. A method as claimed in claim 64 in which the message generation includes wrapping the encrypted data with a symmetric entity confidentiality key which has been securely transferred in advance to the distributor.

70. A method as claimed in claim 64 in which the message generation includes wrapping the encrypted data with the public part of an asymmetric entity confidentiality key pair, the said public part having been securely transferred in advance to the distributor.

71. A method as claimed in claim 69 in which the distributor holds the said public part of the key pair confidential.

72. A method as claimed in any one of claims 1 to 30 in which the client first sends the encryption confidentiality key to the distributor, which uses it to check whether the client is a valid client; the distributor being arranged to send the message only if the client is a valid client.

73. A computer program for carrying out a method as claimed in any one of claims 1 to 30, or as claimed in any one of claims 57 to 72.

74. A machine-readable data-carrier carrying a computer program as claimed in claim 73.

75. A electronic data-stream representative of a computer program as claimed in claim 74.

Seat of Trust Process

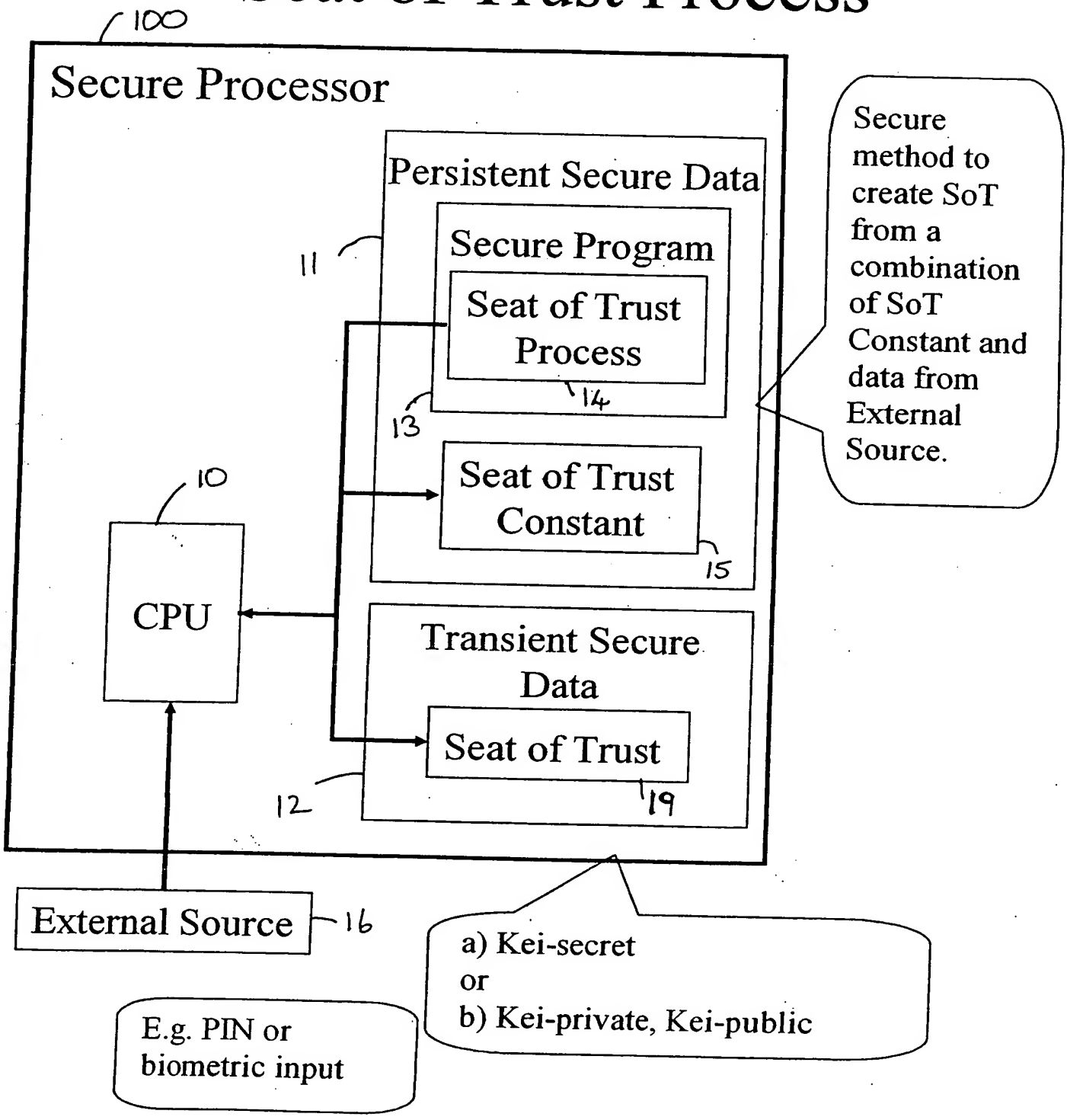


FIG 1
PRIOR ART

Secure Transfer Protocol (1)

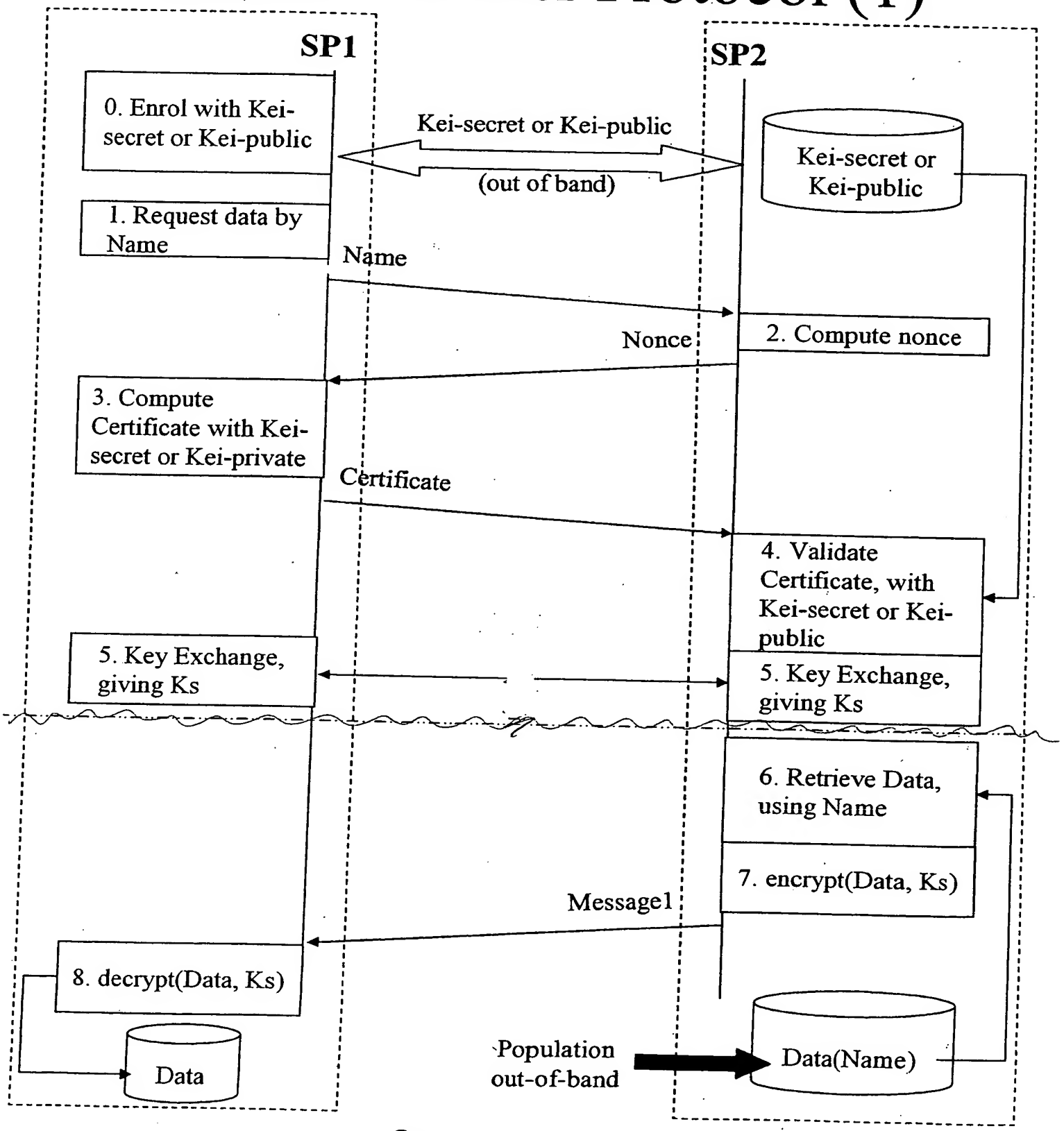


FIG 2
PRIOR ART

Secure Transfer Protocol (2a)

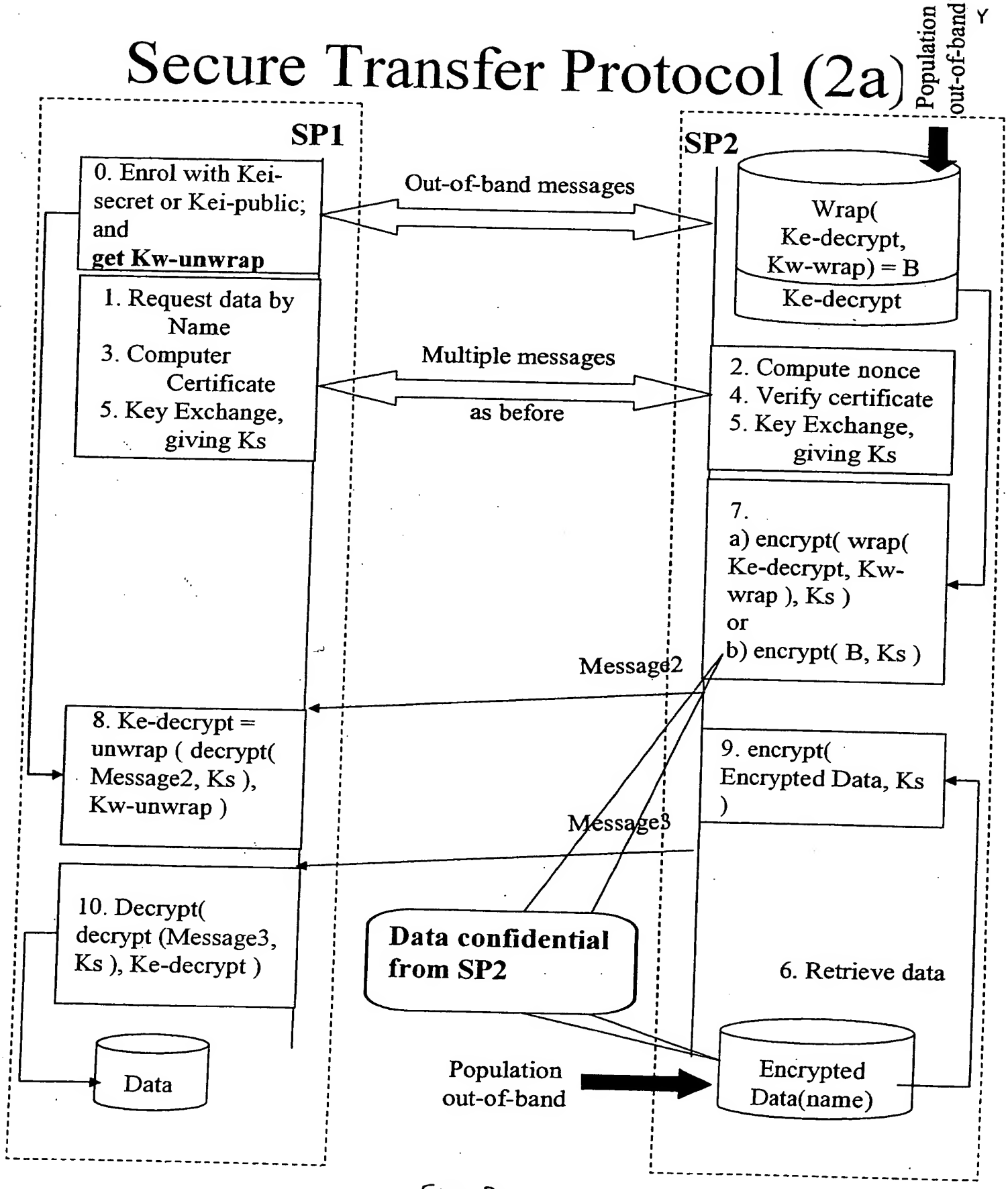


FIG 3

Secure Transfer Protocol (2b)

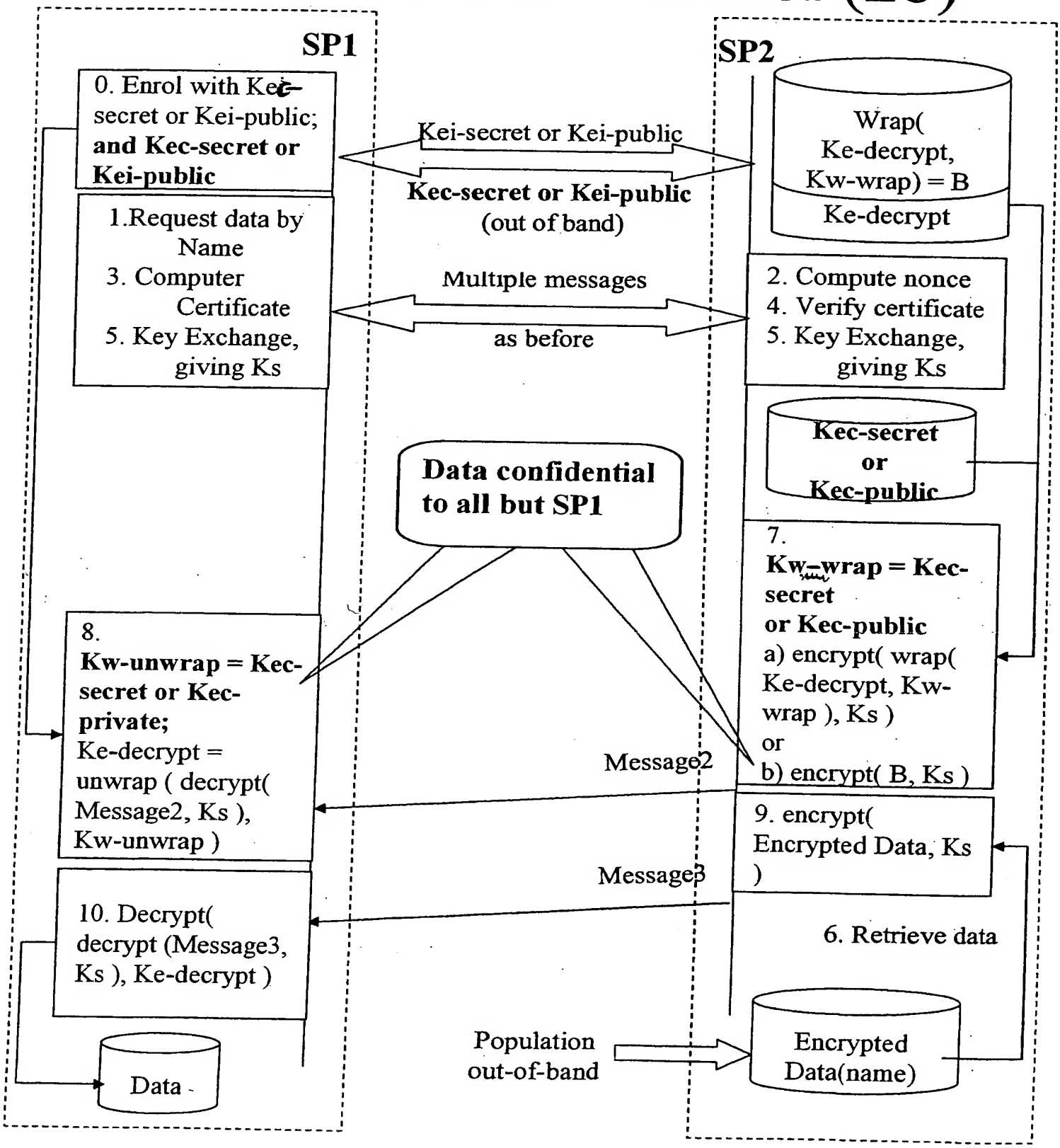


FIG 4

Secure Transfer Protocol (3a)

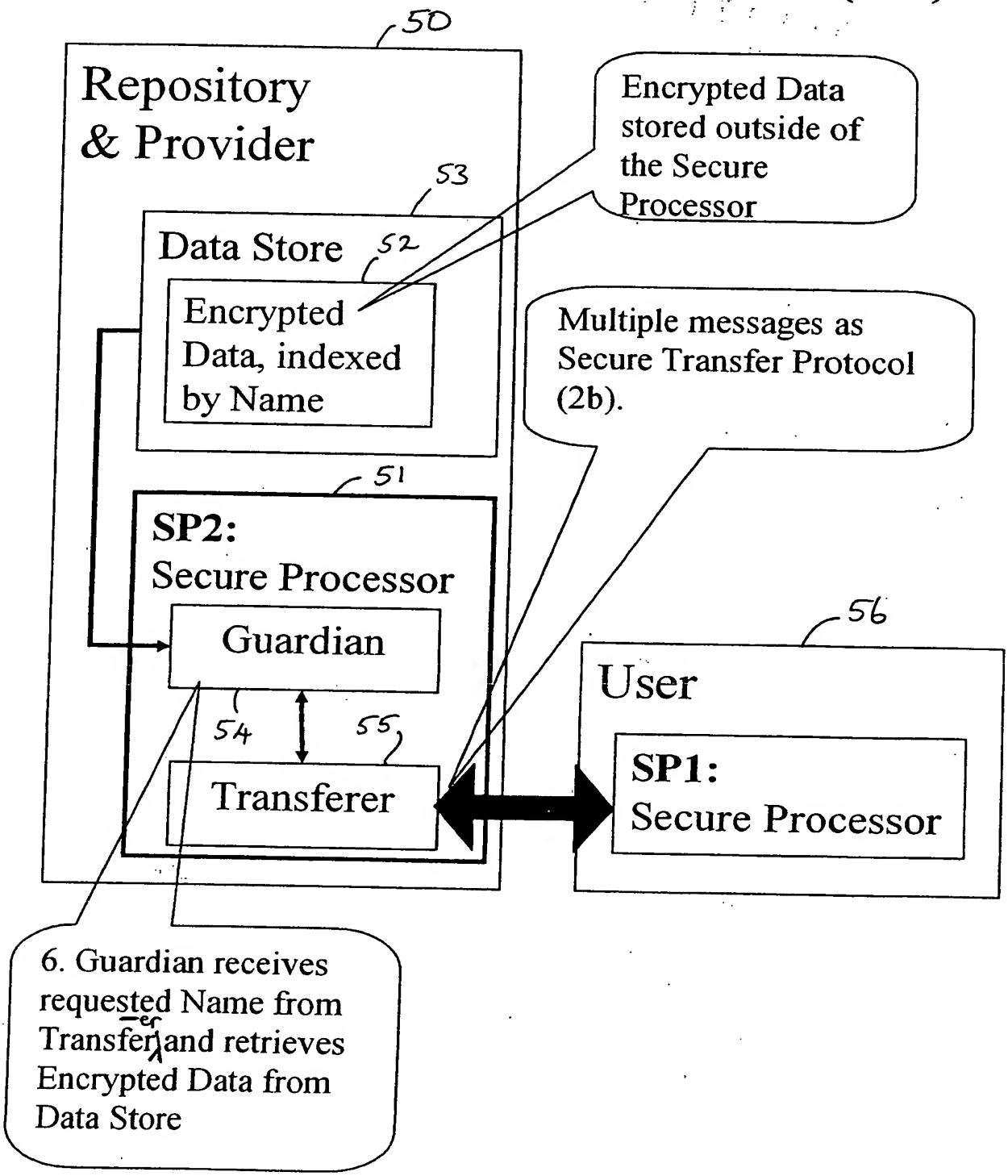


FIG 5

Secure Transfer Protocol (3b)

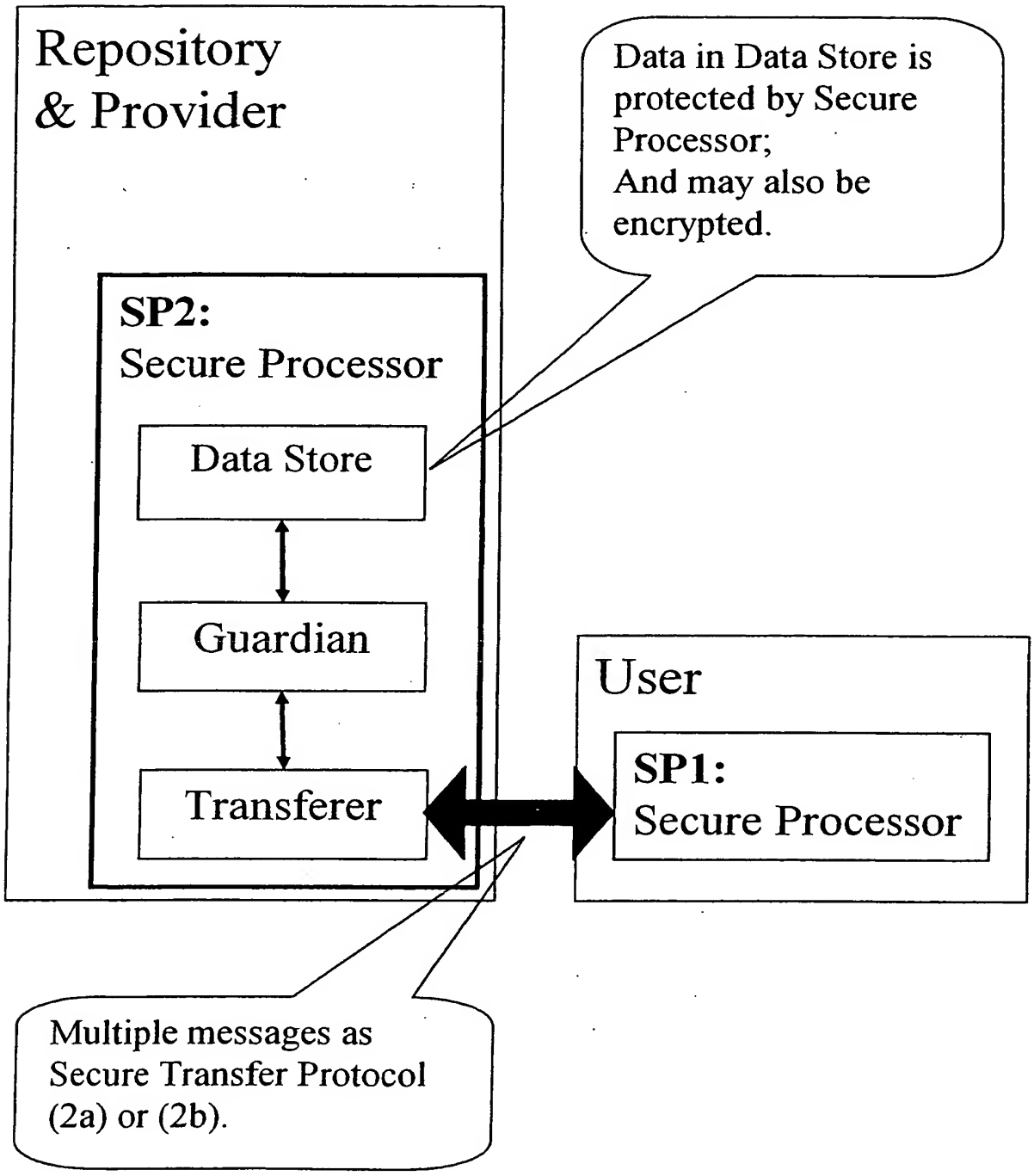


FIG 6

Secure Transfer Protocol (3c)

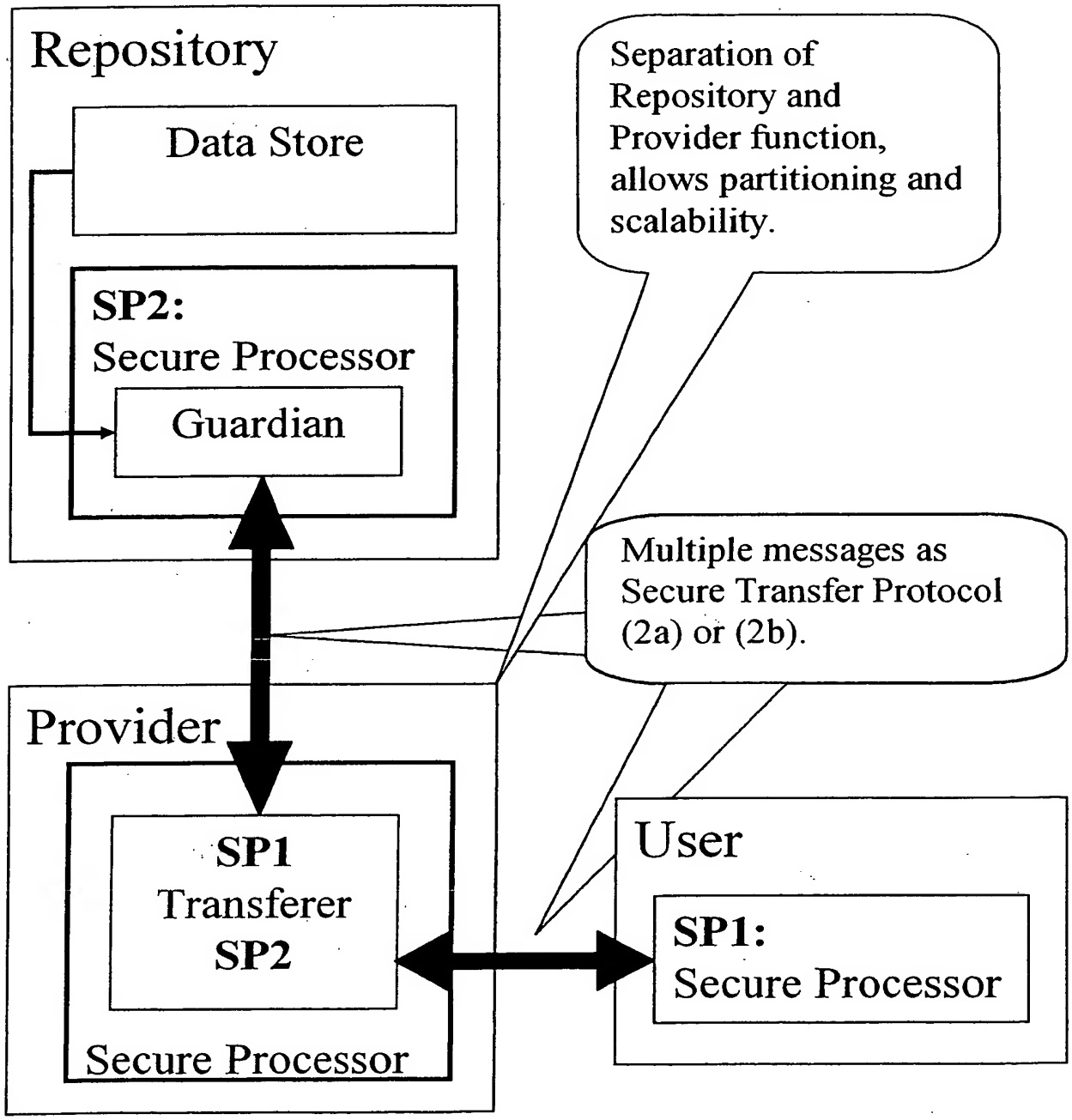


FIG 7

Data Set and Policy

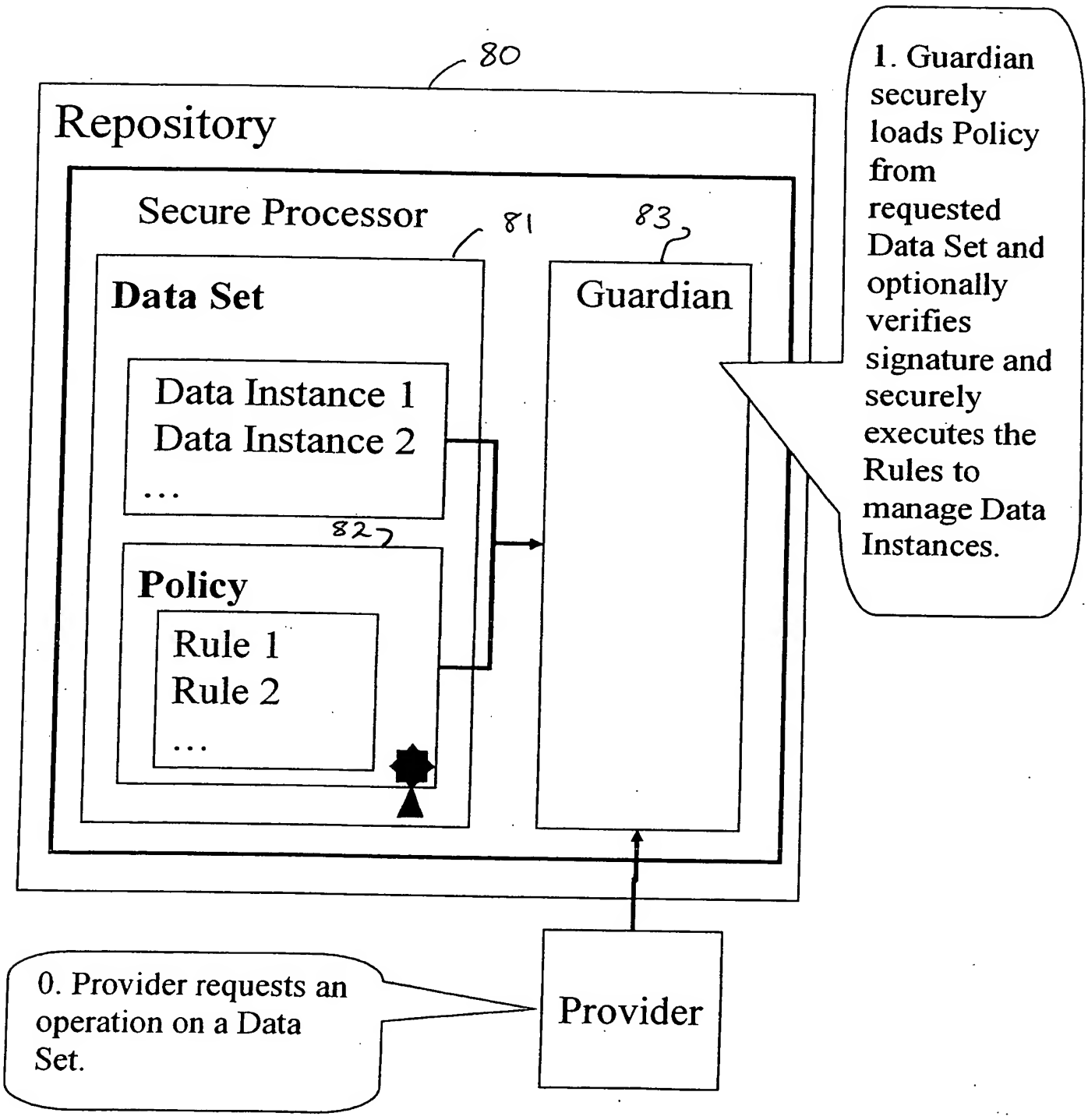
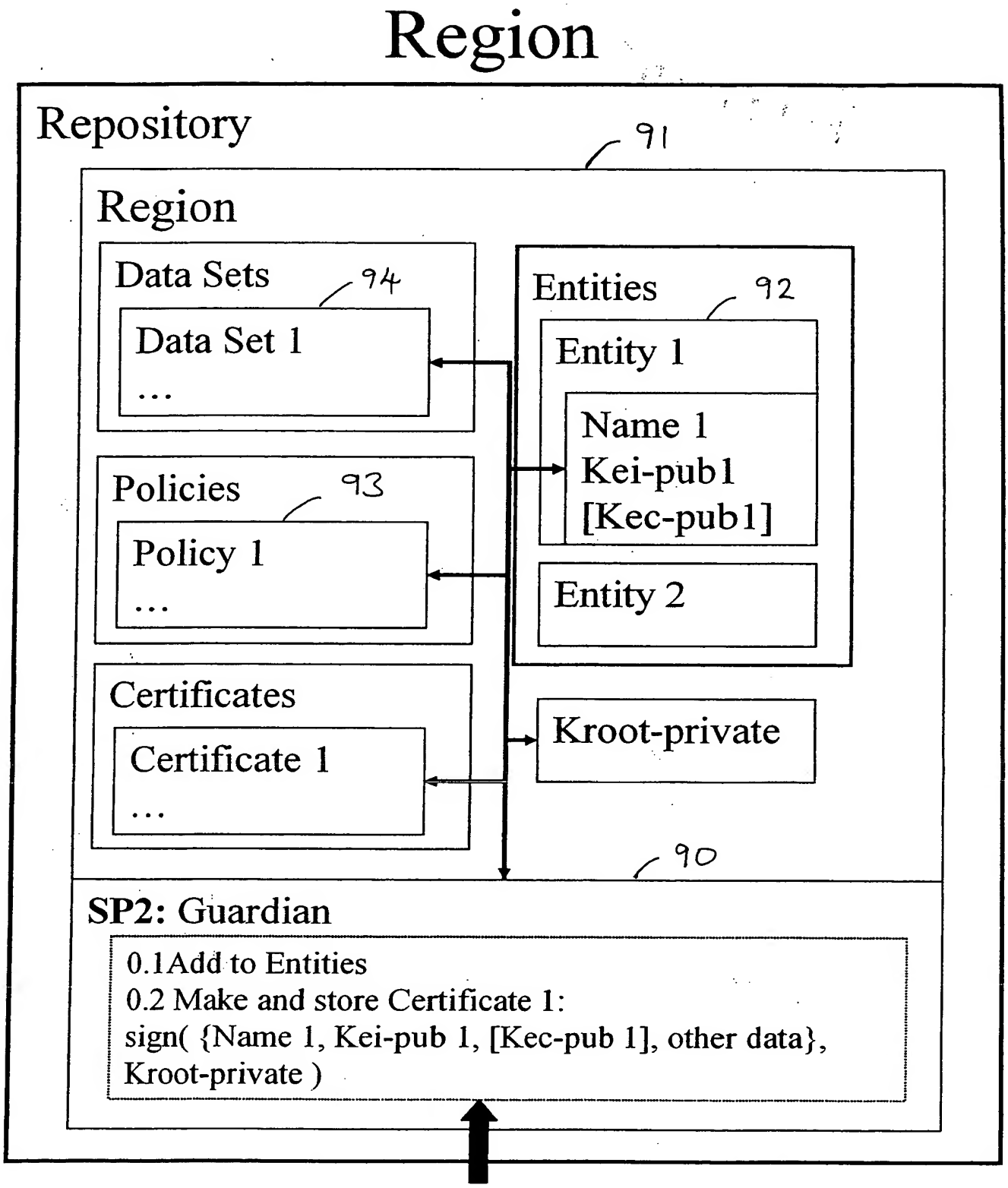


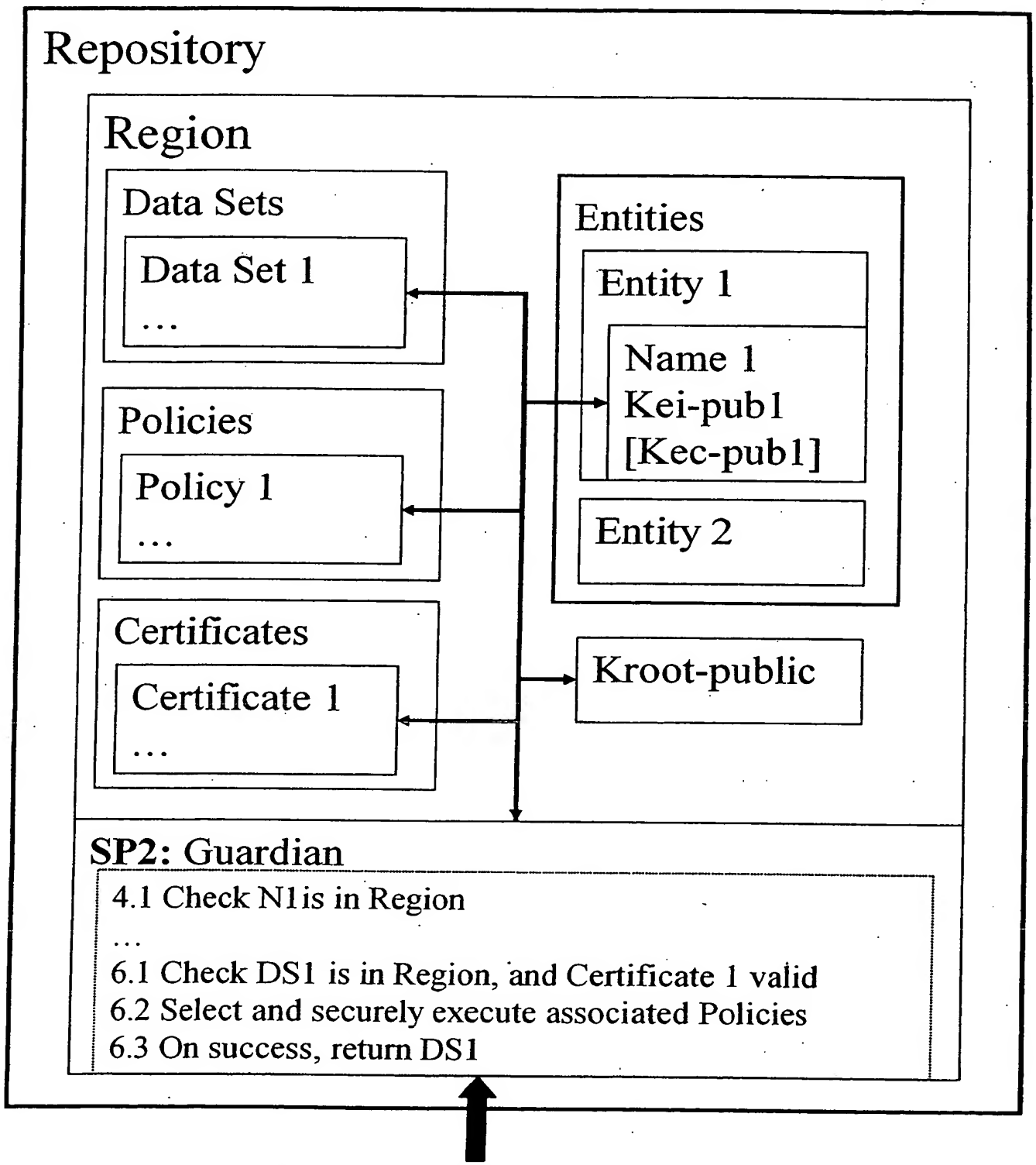
FIG 8



Step 0. Enrol Entity 1
with {Name 1, Kei-pub1} and optionally Kec-pub1

Fig 9

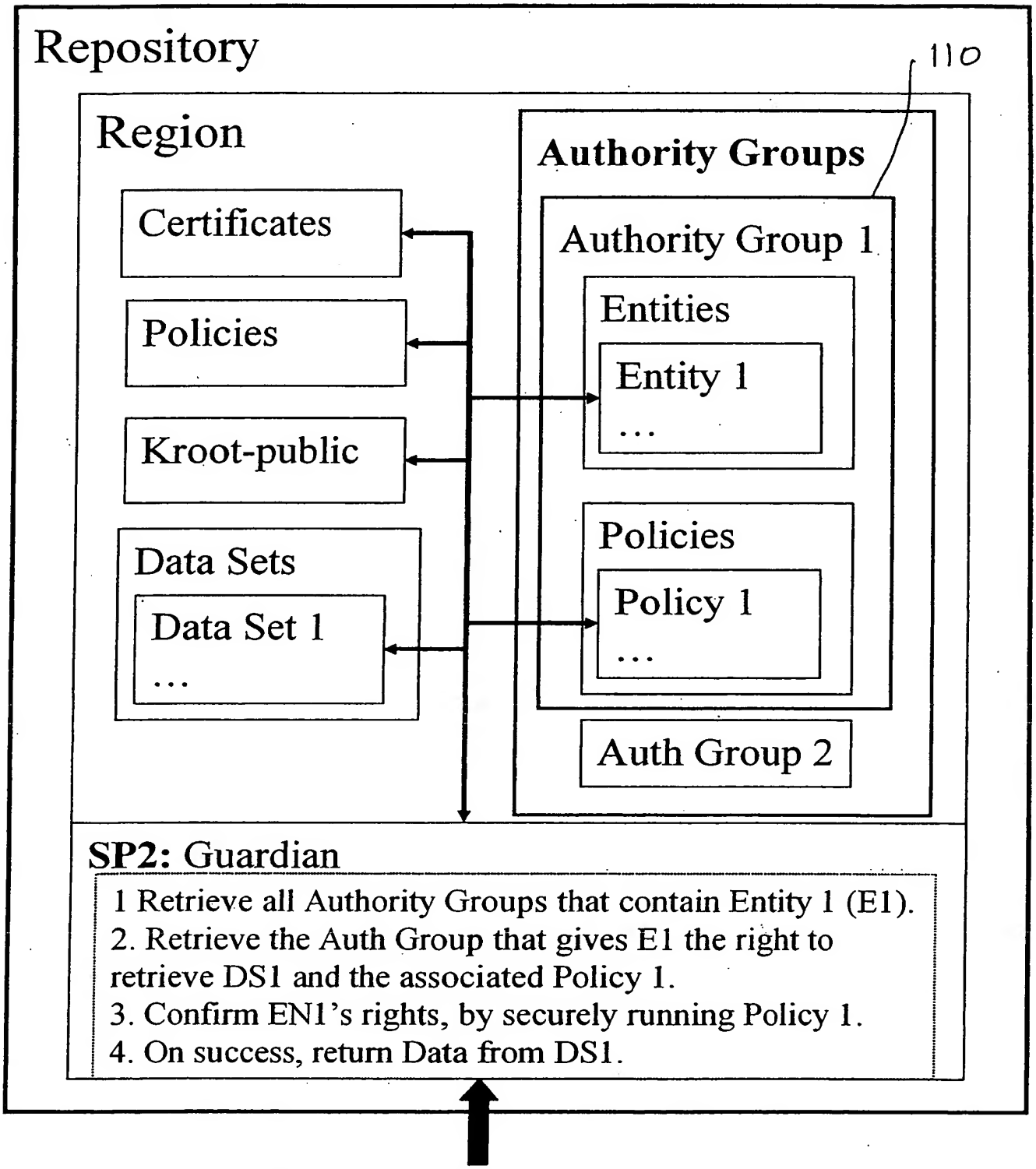
Secure Transfer Protocol (4)



Steps 4. and 6. when retrieving Data
from Data Set 1 (DS1) for Entity, Name 1 (N1)

Fig 10

Authority Group



Request on behalf of Entity 1,
e.g. retrieving Data from Data Set 1 (DS1).

Fig 11

Region Authority Group

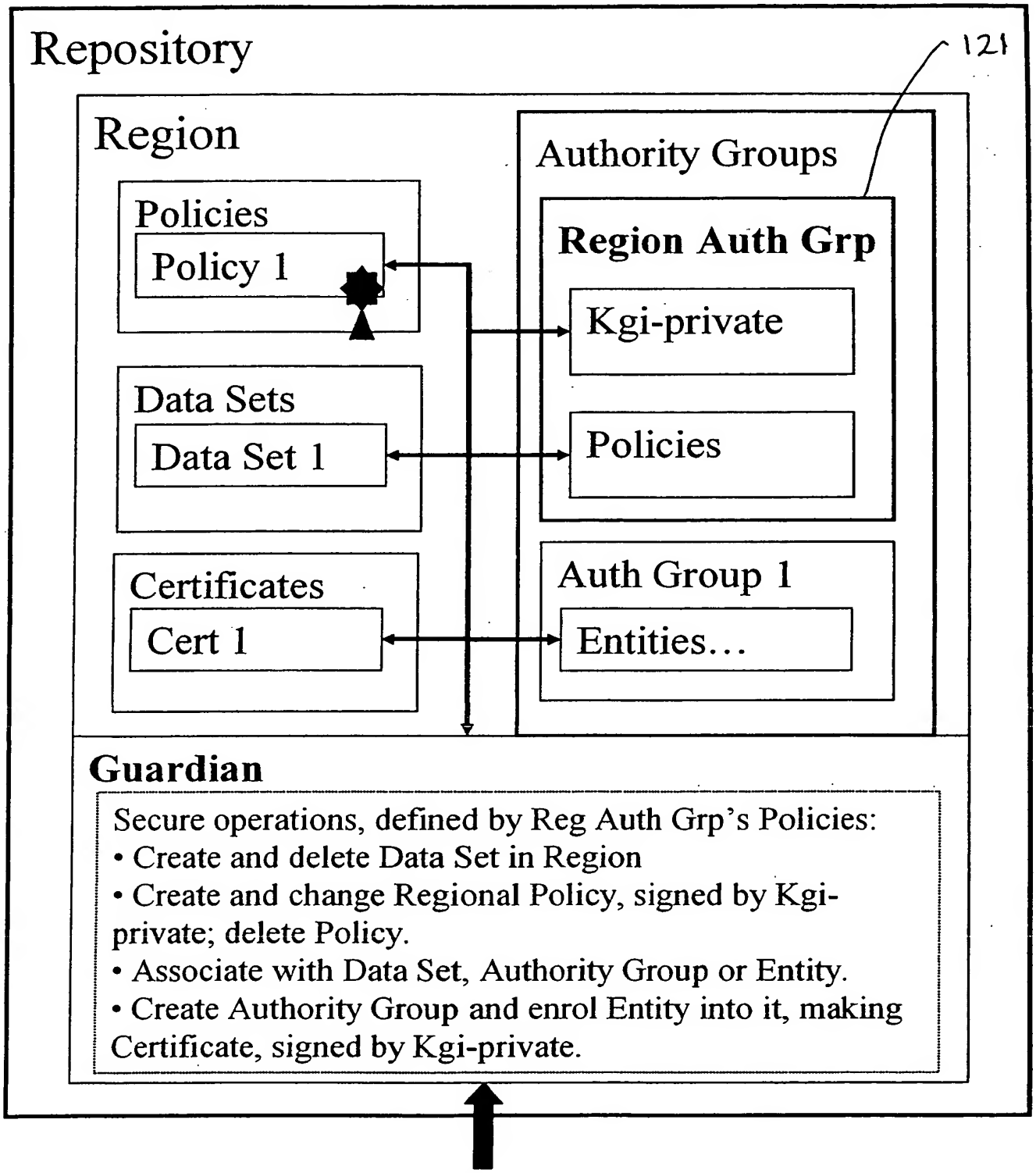


FIG 12

Revoking Entity in Region

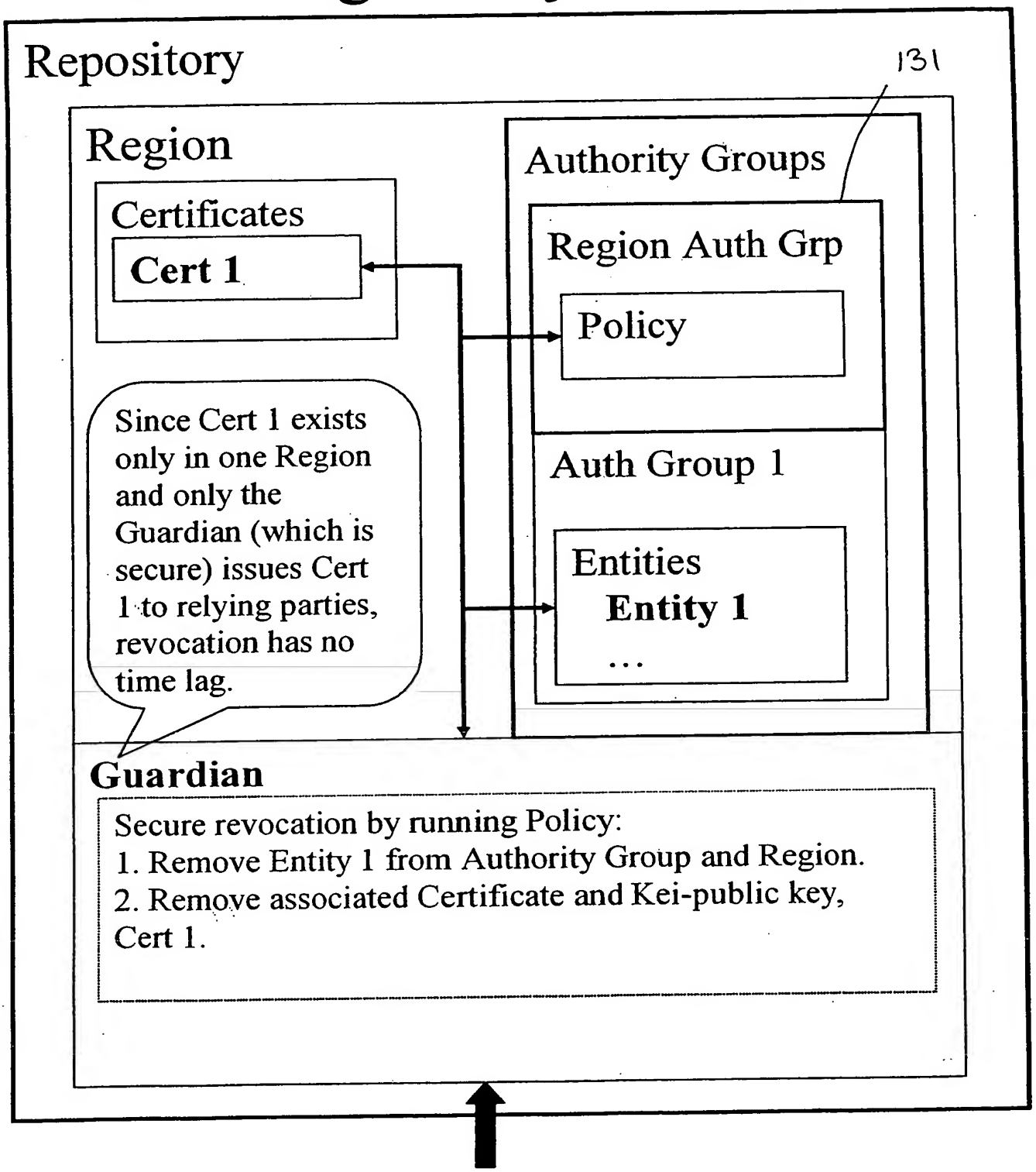
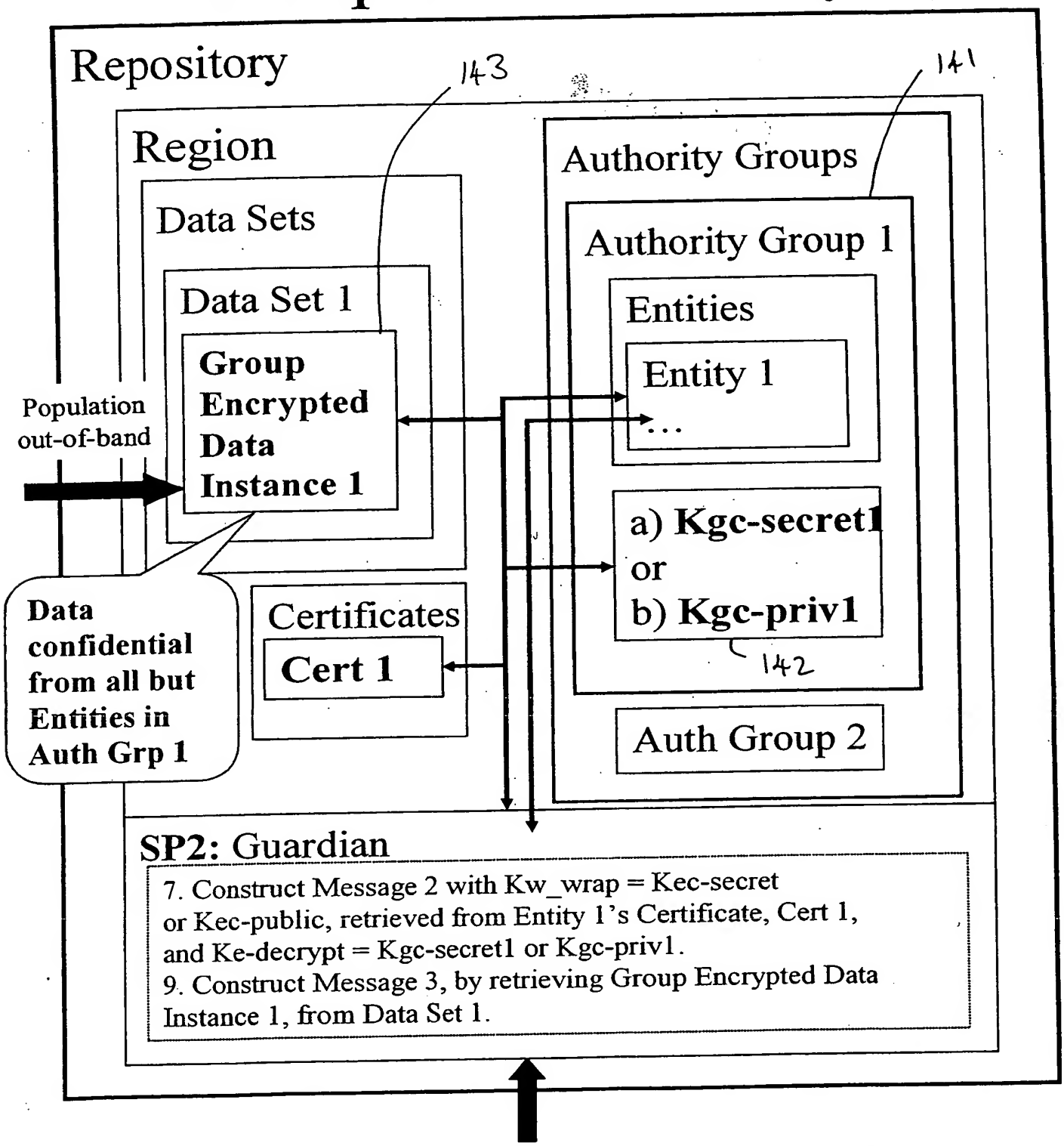


FIG 13

Group Confidentiality



Request to retrieve Group Encrypted Data from Data Set 1.

FIG 14